Three-Tier Satellite Multicast Security Protocol Based on ECMQV and IMC Methods

Attila Altay Yavuz
Computer Engineering Department
Bogazici University, Istanbul, Turkey.
Email:attila.yavuz@boun.edu.tr

Fatih Alagöz
Computer Engineering Department
Bogazici University, Istanbul, Turkey.
Email: alagoz@boun.edu.tr

Emin Anarim
Electrical Engineering Department
Bogazici University, Istanbul, Turkey.
Email: anarim@boun.edu.tr

Abstract— **In this paper, we propose a new three-tier satellite multicast security protocol based on ECMQV (Elliptic Curve Menezes-Qu-Vanstone) and IMC (Improved Merkle Cryptosystem). We make contribution to the satellite multicast security protocols in two major points. These are protocol architecture design and cryptographic methods aspects. Our protocol is specifically designed for multicast systems having very large number of members and highly dynamic member join leave characteristic. Our protocol minimizes the rekeying workload of satellite layers and shows high performance for many criteria using three independent key distribution layers. In addition, our protocol uses a different cryptographic method for each layer and achieves major cryptographic goals together that are not provided in implementations of some other protocols. Using ECMQV and especially IMC based methods in satellite multicast security protocols is a novel approach and has many advantages.**

Index Terms— Multicast Security Protocol, Satellite Communication, ECMQV (Elliptic Curve Menezes-Qu-Vanstone), Layered Protocol Architecture, IMC (Improved Merkle Cryptosystem).

I. Introduction

Satellite multicast applications gain significant importance in today's applications such as pay-per-view programs, multimedia (video and audio conferences), file sharing and specific applications for mobile users. In addition to these, one of the most important application areas of satellite multicast is military command-control. All of these applications require secure and reliable multicast protocols. However, providing security in a satellite multicast system is a difficult task without creating significant performance deterioration. Especially, for a multicast system having very large number of members and highly dynamic member join-leave characteristics (such as mobile environments), providing security causes significant workload on the overall system. In order to provide forward and backward security, the group key must be updated for each member join leave event, in satellite multicast security systems. This requirement creates significant workload on satellites, which are the most resource limited components of satellite multicast systems. Also, cryptographic methods used in satellite multicast security protocols must be selected so that they achieve major cryptographic goals (confidentiality, authentication, integrity and unforgeability) and do not cause significant workload on the multicast system.

In this paper, taking into consideration these problems, we propose a new three-tier satellite multicast security protocol based on ECMQV (Elliptic Curve Menezes-Qu-Vanstone) [1], IMC (Improved Merkle Cryptosystem) [2] and ECPVSS (Elliptic Curve Pintsov-Vanstone Signature Scheme) [3] cryptographic protocols. Our protocol can be applied to very large multicast groups with 10^6 members or more without creating performance and security problems. Our protocol is also suitable for highly dynamic multicast groups having mobile members.

We use three independent key distribution layers to provide scalability and modularity to handle very large multicast systems. The satellite layers of our protocol consists of a GEO satellite layer as the general manager of system and a LEO-MEO satellite layer for bulk data multicast and key distribution purposes. The third layer of our protocol includes terrestrial units (TU), non-trusted servers (NTS) and members. Each layer uses appropriate cryptographic algorithms and key establishment protocols with their alternatives. Using this layered architecture provides many benefits such as decentralization of the multicast and key distribution workloads as well as the centralization of security control.

We use alternative cryptographic methods in our protocol such as ECMQV, IMC and ECPVSS to the naive key exchange and classical RSA and DLP (Discrete Logarithm Problem) based signature approach [4]. The reason is that ECMQV provides efficient and secure collaborative (fair) key exchange between layers when compared to the classical approaches. IMC and IMC based methods are novel approaches and have not been used before in satellite multicast security protocols. ECPVSS provides bandwidth and security advantages in third layer.

Our protocol provides significant advantages for the rekeying workload of the satellite layer, which is one of the most important parameters for satellite multicast systems. It also provides advantages to reduce the number of keys which are stored in satellites and TUs. Apart from these, central security management with decentralized multicast workload is an important advantage.

II. Related Works

Key management protocols use hierarchical approaches to reduce workload of the key management process. These

0-7803-9536-0/06/$20.00 ©2006 IEEE.

approaches can be divided into two major categories [5]: key-based hierarchy and group-based hierarchy. Group-based approaches create hierarchical groups to manage large multicast systems such as [6], [7]. Key based hierarchy approaches use a tree-structured hierarchy for cryptographic keys such as LKH (Logical Key Hierarchy) [8], OFT (One-way Function Tree) [9] and ELK (Efficient Large group Key) [10]. Finally, there are also hybrid approaches such as Mykil [11].

A multicast security protocol, which is specifically designed for satellite multicast systems, is given in [12]. This protocol also uses hybrid approaches. It uses natural hierarchical structure of satellite multicast systems to reduce the workload of the satellite. The basic idea of [12] is providing independency of layers in satellite multicast systems. Consequently, rekeying workload of the satellite can be reduced significantly. Whenever member join-leave event occurs, key update is only done in a local TU or member group and the whole system is not affected from modification. Study in [12] uses LKH protocol in all its layers.

Since we mention Flat and LKH protocols in the following sections, here we give some of their properties. Flat protocol uses a straightforward approach as a key management method. In the Flat system, each member is directly connected to the group manager and a unique key is assigned to a member. Whenever a key update occurs, the group key is sent to each member one by one after being encrypted with unique keys of each member. Thus, the key update cost of the Flat system is N, which is the number of members in the multicast system. LKH protocol, designed for handling moderately large and dynamic groups, uses a logical key tree structure to reduce rekeying cost. In the LKH protocol, each member stores a key vector to reach the group key. This key vector contains the keys which take place on the path of the member to reach the root of the tree. Using this method, for each member join-leave event, only keys that are on the affected paths are updated. This structure reduces rekeying cost of LKH from N to $k \log_k N$, where k is the branching factor of the tree. This provides a significant advantage when compared to the Flat protocol.

III. Cryptographic Techinques Used in Our Protocol

In our protocol, we essentially use elliptic curve cryptography (ECC) based techniques. ECC has many advantages over factorization based [13] and classical DLP [14] based approaches for both computational and key bit length aspect [15]. An application of ECPVSS algorithm to the satellite multicast security can be found in [12]. In this paper, we use ECMQV key agreement protocol [16] as a major cryptographic primitive. Also, as a novel approach, we show that it is possible to use IMC or STAKE (Signcryption type Authentic Key Agreement) alternatively. We give brief descriptions of these protocols and present their advantages.

A. The ECMQV Protocol

The ECMQV protocol is an authenticated key agreement protocol, proposed by Law, Menezes, Qu, Solinas and Vanstone [1], based on the standard authenticated Diffie-Hellmann (DH) key agreement protocol on EC. The ECMQV protocol provides known-key security (KK-S), forward secrecy (FS) and key-compromise impersonation resilience (KCI-R) under the assumption of the intractability of ECDLP (Elliptic Curve DLP). In KK-S, even if sessions are revealed to an adversary, each execution of the key agreement protocol must generate unique and matching session key. In FS, even if long term private keys are revealed to an adversary, the secrecy of previously established session keys should not be affected. In KCI-R, even if one of the instances of communication is corrupted due to private key loss, then an adversary can not masquerade Alice as another principal [17]. Notice that these properties can not be achieved by classical DH based approaches. The ECMQV protocol is also standardized by IEEE P1363. Details can be found in [4]. Moreover, some improvement for MQV, HMQV (Hashed MQV) is also proposed in [18]. It uses hash functions and challenge-based signatures derived from Schnorr Identification Scheme. The study in [19] gives a different approach to HMQV.

The ECMQV protocol is given in Table 1. F is a finite field, E is an elliptic curve, $\#E(F)$ is the number of points on elliptic curve and q is the prime divisor of $\#E(F)$ and also is the order of the curve point G on subgroup of order q. Let x the binary representation of the first coordinate of Q. Let $\overline{Q}$ be defined as $\overline{Q} = x \bmod 2^{\lfloor f/2 \rfloor} + 2^{\lfloor f/2 \rfloor}$ ($Q \leftrightarrow R$ and $\overline{Q} \leftrightarrow \overline{R}$ in Table 1).

TABLE 1
THE ECMQV KEY ESTABLISMENT PROTOCOL

Alice		Bob
1. Alice generates a static private key $w_a \in \{1,\cdots,q-1\}$, computes $W_a = w_a \cdot G$ and Publishes static public W_a	⇳	1. Bob generates a static private key $w_b \in \{1,\cdots,q-1\}$ computes $W_b = w_b \cdot G$ and Publishes static public W_b
2. Alice generates the ephemeral private key $r_a \in \{1,\cdots,q-1\}$, computes corresponding ephemeral public key $R_a = r_a \cdot G$ and sends this value to Bob	⇳	2.Bob generates the ephemeral private key $r_b \in \{1,\cdots,q-1\}$, computes corresponding ephemeral public key $R_b = r_b \cdot G$ and sends this value to Alice
3. Alice computes $s_a = (r_a + \overline{R}_a w_a) \bmod q$ and $R_b + \overline{R}_b W_b = s_b \cdot G$		3.Bob computes $s_b = (r_b + \overline{R}_b w_b) \bmod q$ and $R_a + \overline{R}_a W_a = s_a \cdot G$
4.Alice computes $Z = \frac{\#E(F)}{q} \cdot s_a \cdot (R_b + \overline{R}_b W_b)$		4. Bob computes $Z = \frac{\#E(F)}{q} \cdot s_b \cdot (R_a + \overline{R}_a W_a)$
5. Both Alice and Bob generates shared secret key Z $Z = (\#E(F)/q) \cdot s_a \cdot s_b \cdot G$		

B. The IMC and STAKE

In section III, we have mentioned well-known public key cryptosystems. Generally, key agreement methods including digital signatures are based on these public key cryptosystems.

However, the first cryptosystem, which provides a solution to the secure communication problem over insecure channels without pre-established secrets, is the Merkle Cryptosystem (MC) [20]. In this paper, as an alternative key establishment method, in addition to ECMQV and ECPVSS, we offer to use IMC that has been proposed in [2]. IMC increases the security of the original MC and its variants (VMC) in [21]. In IMC, cryptographic hash functions and a new puzzle structure are used together in order to increase the security of MC and VMC. The key agreement value, which is sent as clear text in VMC, is hidden using a cryptographic hash function in IMC. Also, in order to increase the security of the key agreement value, auxiliary keys are used. In addition to IMC, STAKE protocol is proposed achieving major cryptographic goals such as confidentiality, authentication, integrity and unforgeability together. STAKE is based on IMC and signcryption type key establishment schemes. Most important property of STAKE is that it only relies on symmetric cryptosystem and cryptographic hash functions while achieving major cryptographic goals. IMC based approaches do not need any trusted third party even if it relies on only symmetric cryptosystems and cryptographic hash functions.

IMC and STAKE are novel approaches having these properties and we propose them as alternative methods to use in key establishment steps in a satellite multicast system. These methods are especially suitable for applications that require high security and prefer to use a symmetric cryptosystem. IMC can provide approximately 3000 bit RSA equivalent and 2500 bit sub group DLP equivalent security. However, as an inherent property of MC, IMC and STAKE have non-neglible storage requirements. For this reason, if enough storage is not available then implementing only ECMQV can be more appropriate. However, IMC and STAKE are quite applicable for today's standard hardware systems under usual conditions. Details of STAKE can be found in [2]. We give notations, which are used in IMC:

P : Public key vector (puzzles), K : Secret key vector which is used to generate P. $P_i \in P$, $K_i \in K$ for $1 \leq i \leq N$ where $N = 2^m$. m: The parameter which determines the number of elements in the P and K vectors. $||$ denotes concatenation operation. $(E-D)_K$: Symmetric encryption and decryption functions using secret key K. H : Cryptographic hash function. y_i : Auxiliary secret key which is used to increase bit length security of the hashed message transmitted over the network, P_i^* : Public key which is generated using y_i auxiliary keys. K_{s_a} and K_{s_b} denote session keys, which are generated by Alice and Bob, respectively. h : Secret hashed vector where $h_i \in h$, for all i, $1 \leq i \leq N$ where $N = 2^m$. $PRNG$: Pseudo Random Number Generator.

The brief description of IMC is given below:

1.Alice generates auxiliary secret keys y_i and puzzle pairs $P_i = E_{K_i}(X)$, $P_i^* = E_{K_i}(y_i)$ for $1 \leq i \leq N$ where $N = 2^m$. Alice sends (P_i, P_i^*, X) for all i to Bob and stores (K_i, y_i) pairs as secret key pairs.

2. Alice generates hashed secret key vector h, $h_i = H(K_i||y_i)$ for $1 \leq i \leq N$ where $N = 2^m$. Alice stores h as secret key vector.

3. Bob obtains (P_i, P_i^*, X) for $1 \leq i \leq N$ where $N = 2^m$. Then, he generates random keys l_j such that $while(v, search\ on\ P_i)\{l_j = PRNG(),\ v = E_{l_j}(X),\ move\ indices\}$. If $(P_i == E_{l_j}(X))$ then $K_i = l_j$ and Bob finds one of the secret keys K_i. Using K_i, Bob decrypts P_i^* and obtains secret auxiliary key $y_i = D_{K_i}(P_i^*)$.

4. Bob calculates $h' = H(K_i||y_i)$ and sends h' to Alice. Notice that, only Alice knows K_i and y_i and using these secret key pairs, only Alice can calculate and verify h'. Due to one-way properties of H, Oscar can not find K_i and y_i from h'.

5. Session key agreement can be done in three different ways:

- *Alice determines the session key:* Bob sends h' to Alice. Alice searches h' in vector h. If she finds it then Alice and Bob agree on key $(K_i||y_i)$. Alice generates session key K_{s_s} and calculates $K_{s_s}{}' = E_{K_i||y_i}(K_{s_s})$ and sends $K_{s_s}{}'$ to Bob. Bob decrypts $K_{s_s}{}'$ and obtains $K_{s_s} = D_{K_i||y_i}(K_{s_s}{}')$.Alice and Bob agree on session key K_{s_s}.
- *Bob determines the session key:* Bob generates K_{s_b} and calculates $K_{s_b}{}' = E_{K_i||y_i}(K_{s_b})$. Bob sends $(h', K_{s_b}{}')$ pair to Alice. Alice searches h' in vector h. If she finds then Alice and Bob agree on key $(K_i||y_i)$.Alice decrypts $K_{s_b}{}'$ and obtains $K_{s_b} = D_{K_i||y_i}(K_{s_b}{}')$.Alice and Bob agree on session key K_{s_b}.
- *Alice and Bob jointly determine the session key:* Alice and Bob agree on $(K_i||y_i)$ similar to above and they exchange K_{s_s} and K_{s_b} session keys. They calculate their joint session key $K_s{}' = K_{s_s} \oplus K_{s_b}$.

IV. Architecture and Design Properties of Our Three-Tier Protocol

A. Major Design Properties

We design a new satellite multicast security protocol having three independent LKH layer. Each layer uses appropriate cryptographic algorithms and key establishment protocols together with their alternatives. These cryptographic methods and protocols provide solutions for key establishment between layers. Each protocol is selected taking the properties of each layer into consideration .

Our protocol utilizes the major design principles of the layered architecture, which are the independency and modularity principles. Hybrid key distribution protocols use the divide-and-conquer approach to tackle high rekeying and cryptographic workload of multicast security systems. As we mention in section II, to handle systems having very large number of members and high member join-leave activitiy rate, a combination of key hierarchy and group based hierarchy approaches has been proposed such as [11]. These protocols are designed for general multicast security systems. Applying the independency and modularity principles specifically on satellite multicast systems; the study in [12] provides advantages in terms of both computational effort and rekeying workload.

We use the design principle of [12] in our three-tier approach while improving it in many aspects. We utilize existing layered structure of satellite networks, which is not studied in [12]. LEO, MEO and GEO satellites, already having a hierarchical structure, can be used to design more efficient key distribution protocols. Using different properties of these satellite layers, the overall system performance can be increased and the workload of the individual satellites can be reduced.

B. Architecture of Our Protocol

Our protocol consists of three layers: GEO satellite layer, LEO-MEO satellite layer and terrestrial unit (TU) – member layer. Communication among layers is realized in hierarchical manner taking modularity principle into consideration . In each layer, whenever a member join-leave event occurs, LKH is applied to the related local group. The GEO satellite applies LKH to its LEO-MEO satellite layer. Each LEO-MEO satellite has its own TU group and applies LKH this local TU group. Similarly, each TU has its own member group and applies LKH to this local member group. LKH is preferred like protocol [12] since it has computational and storage advantages for multicast systems as described in section II.

NTS are used to store public key parameters for the ECMQV and IMC protocols. Public keys are transmitted to NTS only once and whenever they are needed, they are obtained from NTS. Consequently, the satellite layer is not affected by future key agreements that will be realized with the same public keys.

The detailed description of our protocol for each layer is given in the next section. Now, we briefly explain properties and responsibilities of each layer.

1) GEO Satellite Layer: GEO satellite layer is responsible for general key management of the overall multicast system. The group keys, which are generated by the GEO satellite, are transmitted to each layer in encrypted form. In this protocol, group keys of the lower layer are known by only the upper layers and keys are only determined by the GEO satellite. This way, the GEO satellite can always control and manage the overall multicast system.

Firstly, the GEO satellite(s) realizes ECMQV key exchange to transmit group keys and seeds to the LEO-MEO satellite layer. Secondly, it generates and transmits group key seeds, which will be used between the LEO-MEO satellite and TUs, to the LEO-MEO satellite layer. Thirdly, it generates and transmits group key seeds that will be used between TUs and members to realize a secure communication. These session keys and seeds should be generated by a CPRNG (Cryptographic PRNG) like Blum-Blum-Shub [22] because they will be used for long term security as a principle of batch keying. Also, as an option, if needed, the GEO satellite may involve data multicast.

2) LEO-MEO Satellite Layer: This layer is mainly responsible for bulk data multicast to TUs and distributing group keys to the TUs and NTS. LEO-MEO satellites obtain shared secret keys from the GEO satellite. Each LEO-MEO satellite uses these shared secret keys to obtain the related group key to communicate with the upper layer. The group key seeds are used to generate group keys. Then, they distribute the group keys and session key seeds to the TU using either ECMQV or IMC-STAKE protocols also involving NTS if necessary.

3) TU-Member Layer: In this layer, TUs are responsible for decrypting the data coming from the LEO-MEO satellite layer and multicasting it after encrypting it with required group keys. TUs obtain and use group keys to realize secure communication with the LEO-MEO satellite layer using either ECMQV or IMC-STAKE protocols also involving NTS if necessary. TUs use seeds to generate group keys that will be used for secure bulk data multicast. Figure 1 shows the architecture of our protocol.

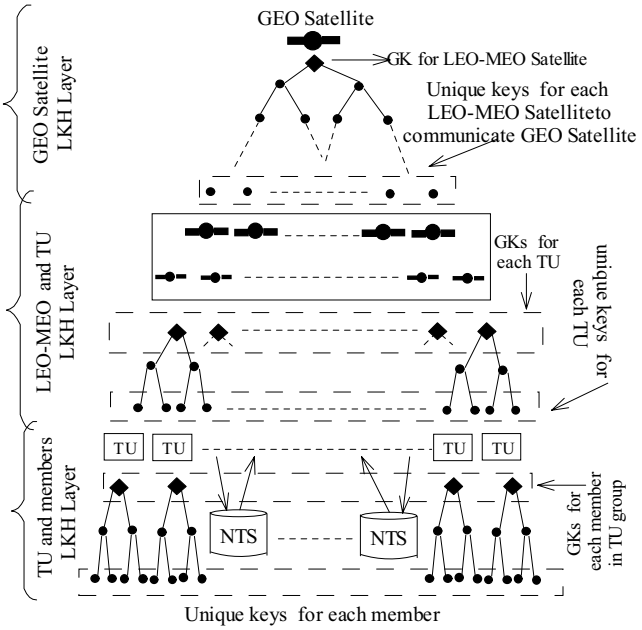

Fig. 1 Architecture of the our satellite security protocol

V. Details of Our Three Tier Protocol

In each layer, there is a hierarchical key exchange and transmission from top to down. The GEO satellite is responsible for generating and distributing seed values $s\alpha_i$ and $s\beta_i$. These seed values are used to generate group key vectors α_i and β_i. Each $s\alpha_i$ seed value is assigned to a LEO-MEO satellite by the GEO satellite. Each LEO-MEO satellite generates group key vector α_i using seed $s\alpha_i$. Elements of group key vector α_i are $\alpha_{i,j}$ where $j \geq l$. So, $\alpha_{i,j}$denotes j^{th} group key which is used by i^{th} satellite in LEO-MEO satellite layer. Each satellite realizes bulk data multicast using these group keys $\alpha_{i,j}$ to their local TU groups and also transmits seed values $s\beta_i$ to the related TUs.

Each TU realizes the ECMQV key exchange with the upper layer to obtain and transmit the required keys. Each $s\beta_i$ seed value is assigned to a TU. Each TU generates the group key vector β_i using seed $s\beta_i$. Elements of group key vector β_i are $\beta_{i,j}$ where $j \geq n_l$. $\beta_{i,j}$ denotes j^{th} group key for i^{th} TU in TU-Member layer. TUs use group keys $\alpha_{i,j}$ to decrypt multicast

data and seed values $s\beta_i$. Using ECPVSS, different from upper layers, TUs transmit $\beta_{i,j}$ keys to members and realize bulk data multicast using these group keys $\alpha_{i,j}$. Members, using ECPVSS, obtain group keys and decrypted bulk multicast data securely. ECPVSS is preferred in this layer because "fair key exchange" may not be preferred in TU-Member hierarchy. Thus, only key transport is realized by ECPVSS like [12].

Seed vectors $s\alpha$ and $s\beta$ are used for batch keying: Thus, instead of sending group key vector elements in α_i and β_i one by one, only their seed values are transmitted. LEO-MEO satellites generate group keys using these seeds and important bandwidth and rekeying cost advantages are gained. Details of each step for layers are given in the following part, where:

n_s : Number of satellites in LEO-MEO satellite layer. l : Number of TUs in TU-Member layer. N : Total number of members in multicast system. $n_l \approx N/l$:Average number of members in one TU local group. $ECMQVKG$: Static Key Generation for ECMQV protocol.

A. GEO Satellite Layer

The GEO satellite generates required group keys and seeds for overall system. Also, if it is necessary, the GEO satellite may realize bulk data multicast.

A.1) GEO satellite generates the group key seed:

$s\alpha_i = CPRNG(\geq n_s)$, $s\beta_i = CPRNG(\geq l)$.

A.2) GEO satellite generates static public-private key pairs from EC curve E and validates them. ECMQV key exchange is done using public keys of lower layer W_{b_i} and required private keys. Using these, GEO satellite has shared secret keys (unique keys) with each satellite at the lower layer:

$(W_{a_i}, w_{a_i}) = ECMQVKG(E, n_s)$, $NTS_i \leftarrow W_{a_i}$ and $W_{b_i} \leftarrow NTS_i$, $Z_i = ECMQV(W_{b_s}, w_{a_i}$, required private keys) unique keys for each satellite in LEO-MEO satellite layer.

A.3) GEO satellite generates and distributes group key GK to the all satellites in LEO-MEO satellite layer using secret shared key Z_i. This group key is used to transmit group key seeds and bulk data multicast if necessary:

$GK_i' = E_{Z_i}(GK)$, for each satellite in LEO-MEO layer.

A.4) GEO satellite multicast seed vectors $s\alpha$ and $s\beta$ using GK. Vector $s\alpha$ is used to generate group keys $\alpha_{i,j}$ that LEO-MEO satellites will use to communicate with TUs. Vector $s\beta$ is used in TU-Member layers. Optionally, bulk data multicast can be realized.

A.5) During each member join-leave, LKH protocol is applied to the LEO-MEO satellite layer using GK to update the required keys.

Optionally, If IMC or STAKE is used; GEO satellite(s) generates seed vectors st, sx and transmits them to the LEO-MEO satellite layer. These seed vectors are used to generate private-public keys for IMC based cryptosystem:

$st_i' = E_{GK}(st_i)$, $sx_i' = E_{GK}(sx_i)$.

B. LEO-MEO Satellite Layer

LEO-MEO satellites, using satellite inter-networking, determines which TUs are managed by which satellite. Group key and group key seed distribution are done according to this agreement.

B.1) LEO-MEO satellites generate static public-private key pairs from EC curve E and validate them. ECMQV key exchange is done using public keys of upper and lower layers using required private keys. Z_i secret keys are shared with the GEO satellite and Z_i' secret keys are shared with TUs. NTSs are used to store and obtain public keys. Using Z_i, each satellite obtains group key GK:

$(W_{b_i}, w_{b_i}) = ECMQVKG(E, n_s)$, $NTS_i \leftarrow W_{b_i}$ and $W_{a_i} \leftarrow NTS_i$, $Z_i = ECMQV(W_{a_i}, w_{b_i}$,required private keys).

$GK = D_{Z_i}(GK_i')$ realized for each LEO-MEO satellite.

$(W_{b_i}', w_{b_i}') = ECMQVKG(E, l)$, $NTS_i \leftarrow W_{b_i}'$ and $W_{t_i} \leftarrow NTS_i$, $Z_i' = ECMQV(W_{t_i}, w_{b_i}'$;required private keys) realized for each TU.

B.2) Using GK, LEO-MEO satellites obtain group key seeds $s\alpha$ and $s\beta$. Each satellite uses its seed $s\alpha_i$ to generate vector α_i. Suppose that, i^{th} satellite uses group key $\alpha_{i,j}$ at the current state. After that, whenever a key update occurs, the i^{th} satellite uses next group key such that $\alpha_{i,j} \rightarrow \alpha_{i,j+1}$. This group key is used to manage the TU group which the satellite is responsible for.

$s\alpha_i = D_{GK}(s\alpha_i')$, $s\beta_i = D_{GK}(s\beta_i')$. If the GEO satellite sends message, $M = D_{GK}(M')$. Using $s\alpha_i$, sufficient number $\alpha_{i,j} = CPRNG(s\alpha_i, \geq l)$ group key is generated $(j \geq l)$.

B.3) Each LEO-MEO satellite sends group key to the TUs that are responsible for using shared secret key Z_i'. Also, seeds $s\beta_i$ are transmitted to the related TUs:

$\alpha_{i,j}' = E_{Z_i'}(\alpha_{i,j})$, $s\beta_i' = E_{Z_i'}(s\beta_i)$.

B.4) Bulk data multicast is done to the TUs using group keys: $M' = E_{\alpha_{i,j}}(M)$.

B.5) Each LEO-MEO satellite applies LKH its local TU group independently for each member join-leave event.

Optionally, If IMC or STAKE is used, only MEO satellites obtain seed vectors st, sx and generate IMC secret key vector K using st. Then using K and sx, satellite generates public key vectors and transmits them to the NTSs. Only MEO satellites are involved in this process since they have higher computational and bandwidth resources.

C. TU-Member Layer

TUs are responsible for realizing the bulk data multicast to the members. Each TU belongs to a TU group, which is managed by a LEO or MEO satellite. Also, each TU has a large member group.

C.1) Like the upper layer, TUs generate static public-private key pairs from EC curve E and validate them. ECMQV key exchange is done using public keys of the upper layer using the required private keys. Z_i' secret keys are shared with the LEO-MEO satellite layer. NTSs are used to store and obtain public keys. Note that TU sends group keys to the members using ECPVSS, which has advantages if collaborative (fair) key exchange is not needed. The following steps are performed for TU and members:

$(W_{t_i}, w_{t_i}) = ECMQVKG(E, l)$, $NTS_i \leftarrow W_{t_i}$ and $W_{b_i}' \leftarrow NTS_i$, $Z_i' = ECMQV(W_{b_i}', w_{t_i}$,required private keys).

$\alpha_{i,j} = D_{Z_i'}(\alpha_{i,j}')$ and $s\beta_i = D_{Z_i'}(s\beta_i')$ realized by LEO-MEO satellite. Each TU generates group keys from seeds $s\beta_i$, $\beta_{i,j} = CPRNG(s\beta_i, \geq n_l)$, $j \geq n_l$. Each TU decrypts the bulk multicast data using group key from which they obtain LEO-MEO satellite. Then, TUs realize bulk data multicast to the members using group keys:

$M = D_{\alpha_{i,j}}(M')$ and $M' = E_{\beta_{i,j}}(M)$. Each TU transmits group key $\beta_{i,j}$ to the members by using $NTS_i \leftarrow$Public parameters for ECPVSS.

C.2) Members obtain group keys and decrypt bulk multicast data securely:

Each member obtains group key $\beta_{i,j}$ from TU. (Public key parameters for ECPVSS)$\leftarrow$ NTS_i, $\beta_{i,j} = ECPVSS\_Unsign(\beta_{i,j}, PVSS\_Parameters)$. Members decrypt multicast data $M = D_{\beta_{i,j}}(M')$.

VI. Performance Comparison and Results

In this paper, two aspects of satellite multicast security protocols are analyzed and improved. Firstly, appropriate cryptographic methods are selected and integrated to our satellite multicast security protocol. Secondly, architecture and design properties of the satellite multicast security protocol are improved. We firstly focus on the performance gain and advantages of our protocol for its design and architecture aspects. Then, we give advantages of cryptographic methods, which are used in our protocol when compared to classical approaches.

A. Performance Comparison of Architectural and Design Aspects of Protocols

Table II shows performance comparison of our protocol to Flat, LKH and our previous protocol given in [12]. The comparison is based on five major criteria: Rekeying workload for satellite layer and TU, number of keys stored in satellite layer, in TUs and members on the average.

The most important criterion is the average rekeying workload of the satellite layer. Since the most resource limited part of the satellite multicast system is the satellite layer, we aim to minimize rekeying workload of this part. For this criterion, among the compared protocols, the most efficient one is our proposed protocol: In Flat and LKH, for each member join-leave event, the key update cost is N and $k \log_k N$ respectively. Notice that, the major parameter that determines the rekeying workload of the system is the rekeying factor r, which is the total number of member join-leave events for a certain time period. In Flat and LKH, since the satellite directly controls all members, a rekeying occurs for each member join-leave event, according to the key update rule of the protocol. Thus, total rekeying workload for Flat and LKH are $N \cdot r$ and $(k \log_k N) \cdot r$ respectively. For the protocol in [12], satellite layer is not affected from member join-leave events due to independency of layers principle. Thus, rekeying of workload of the satellite layer in our previous protocol is $k \log_k l / m_1$ where l is the number of TUs in the related local TU group and m_1 is the batch keying factor. In our proposed protocol, rekeying workload of the LEO-MEO satellite determines the average rekeying workload for all satellite layers. q is the average number of TUs, which are controlled by a single LEO or MEO satellite. Thus, rekeying only occurs for a satellite if one of the TUs is down or violates the security policy. Notice that, these events occur rarely. Also, the batch keying factor becomes $m_2 > m_1$ since only seed values are transmitted. This makes possible to realize more batch keying. In addition to this, parameter m_2 reflects the advantages of lower packet loss and propagation delay values of LEO-MEO satellites when compared to our previous protocol. Thus, the rekeying factor of LEO-MEO satellite layer is $(k \log_k q)/m_2$. Notice that the independency principle is also valid for our proposed protocol and rekeying factor r does not affect the satellite layer. The rekeying workload of the GEO satellite is small and neglible: $k \log_k n_s \leq 10$. As a result, since $q << l$ and $m_2 > m_1$, our proposed protocol is more efficient than our previous protocol.

For average number of keys stored in satellite layer criterion, Flat and LKH require N unique keys for each member. In our previous protocol, the satellite only contacts with TUs and the storage requirement is l. In the proposed protocol, in order to obtain performance gain for the rekeying workload, we slightly increase the number of keys stored in the satellite layer. Notice that the major parameter that determines average number of keys stored in the satellite layer is the storage requirement of the GEO satellite. In ECMQV protocol, key pairs $W_{a_i}, w_{a_i}, r_{a_i}, R_{a_i}$ Z_i and seed values $s\alpha_i$ and $s\beta_i$ are stored in GEO satellite. Thus, the average number of keys stored in the satellite layer is $6n_s + l + 1 \approx 2l$. The number of keys stored in the LEO or MEO satellite is small and neglible: $q \leq l$.

Rekeying workload of TUs is only analyzed for previous and our proposed protocol. For Flat and pure LKH, TUs are not specifically mentioned and are not considered in this comparison. Rekeying workload of the TU is the same for both our previous and proposed protocols, that is $\log_k n_l$. The reason is that, in both protocol, TUs are only responsible for their own local member group for rekeying processes. Also, the number of keys that TU and a member stores are the same for both protocols and is $n_l + \log_k l$ and $\log_k l$, respectively. The number of keys that a member stores is $\log_k N$ in LKH and one in Flat (directly communicates with satellite).

B. Advantages of Cryptographic Methods Used in Our Proposed Protocol

In our proposed protocol, ECPVSS, ECMQV and alternatively IMC-STAKE algorithms are used. Table III shows a comparison of these algorithms and other prevalently used approaches in key exchange methods. DH-ECDH protocols are frequently used for key exchange. However, pure implementation of these protocols is insecure. Man-in-the-middle attack is the most well-known attack used againts these protocols. Also, these protocols do not provide KK-S, FS and KCI-R security

properties. Notice that for signature variants and ECPVSS; KK-S, FS and KCI-R are not compared because signature variants and ECPVSS are not key exchange protocols.

In table III, taking into consideration the properties of algorithms, it is shown whether the algorithm provides the mentioned property (authentication, integrity, unforgeability) or not. Also, for three criteria, bandwidth efficiency, computational effort and confidentiality, VL (very low), L (low), M (moderate), H (High) and VH (very high) levels are assigned. These assignments are done comparing algorithms with each other for their computational efforts, storage requirements and provided security (equivalent bit length security). Bandwidth efficiency assignment is done according to the required packet bit length to realize a cryptographic method. Computational efforts are assigned according to the computational complexity of algorithms and various implementation considerations. Confidentiality comparison is done according to the cryptoanalysis properties of algorithms. For different metrics, comparison and properties of these protocols can be found in [1], [3], [15], [23], [24]. For instance, signature variants provide authentication, integrity and unforgeability but they require transmission of signature together with the message. For small messages, this situation causes significant bandwidth consumption. ECPVSS is a message recovery type signature and solves this problem efficiently. IMC-STAKE also achieves major cryptographic primitives and provides high security. However, it is not appropriate for bandwidth constraint applications.

VII. Conclusion

In this paper, we propose a new satellite multicast security protocol that provides many advantages compared to some well-known multicast security protocols and as well as our previous protocol. Using three independent LKH layers provides significant performance gain for rekeying workload of the satellite layer and reduces the number of keys that are stored in satellites. Our protocol makes possible the centralization of security management and the decentralization of multicast workload at the same time. Moreover, using LEO-MEO satellite inter-networking, the delay problem is minimized and the workload of the individual satellite is reduced. Another benefit of our protocol is that it increases batch keying factor. Using ECMQV in the satellite layer provides “fair key exchange”. It also achieves many cryptographic primitives, which can not be achieved by some other protocols. IMC and STAKE methods are newly proposed and have not yet been used in satellite multicast security protocols providing some advantages. Also, advantages of ECPVSS in our previous protocol remain the same in the third layer of our protocol. Advantages of our proposed protocol to the Flat, LKH and our previous protocol can be observed in Table II and Table III.

ACKNOWLEDGEMENTS

This work is supported by the State Planning Organization of Turkey under “Next Generation Satellite Networks Project”, and Bogaziçi University Research Affairs.

References

[1] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. An efficient protocol for authenticated key agreement. Designs, Codes and Cryptography, pages 28:119–134, 2003.

[2] A. Altay Yavuz, E. Anarim, F. Alagoz. Improving Merkle cryptosystem and desinging signcryption type authentic key establishment protocol, under preparation, 2006.

[3] IEEE P1363a/D2. Standart specifications for public key cryptography: Pintsov-Vanstone Signature with message recovery, January 10, 2000.

[4] Standard specifications for public key cryptography. IEEE P1363/D13, November 1999.

[5] S. Mishra. Key management in large group multicast. Technical Report CU-CS-940-02, Department of Computer Science, University of Colorado, Boulder, CO., 2002.

[6] S. Mittra. Iolus: A framework for scalable secure multicasting. InProceedings of the ACM SIGCOMM’97, September 1997.

[7] S. Setia, S. Koussih, and S. Jajodia. Kronos: A scalable group re-keying approach for secure multicast. In Proceedings of the IEEE Symposium on Research in Security and Privacy, May 2000.

[8] D. Wallner, E. Harder, and R. Agee, Key management for multicast: Issues and architectures, IETF, RFC2627, June 1999.

[9] D. Balenson et al. Key management for large dynamic groups: One-way function trees and amortized initialization, IETF Draft, work-in-progress, draft-balenson-groupkeymgmt-oft-00.txt, February 1999.

[10] A. Perrig, D. Song, and J.D. Tygar. ELK, a new protocol for Efficient Large-group Key distribution. IEEE Security and Priavcy Symposium May 2001.

[11] J. Huang and S. Mishra. Mykil: A Highly scalable and efficient key distribution protocol for large group multicast. In the IEEE 2003 Global Communications Conference (GLOBECOM 2003), San Francisco, CA (December 2003).

[12] A. Altay Yavuz, F. Alagoz, E. Anarim. A new satellite multicast security protocol based on elliptic curve signatures. 2nd IEEE International Conference on Information & Communication Technologies(ICTTA) , April 2006.

[13] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM, 21(1978), 120-126.

[14] Taher ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, v. IT-31, n. 4, 1985, pp469–472 or CRYPTO 84, pp10–18, Springer-Verlag.

[15] Kristin Lauter. The Advantages of elliptic curve cryptography for wireless security. IEEE Wireless Communications, February 2004.

[16] N. Smart and P. Leadbitter. Analysis of the insecurity of the ECMQV with partially known nonces. In Proceedings ISC 2003, pages 240–251. Springer-Verlag LNCS 2851, August 2003.

[17] M. Adriano Strangio, Efficient Diffie-Hellmann two-party key agreement protocols based on elliptic curves, Proceedings of the 2005 ACM symposium on Applied computing, Pages: 324 - 331, 2005.

[18] H. Krawczyk, HMQV: A high-performance secure Diffie-Hellman protocol", Advances in Cryptology, CRYPTO 2005, Lecture Notes in Computer Science, 3621 (2005), 546-566.

[19] A. Menezes, Another look at HMQV, In http://eprint.iacr.org/2005/205, June 27, 2005.

[20] C. Merkle. Secure communications over insecure channels, Communications of the ACM 21(4), pp294–299 (April 1978).

[21] Chris Mitchell. Public key encryption using block ciphers, technical report RHUL-MA-2003-6, 9 September.

[22] Lenore Blum, Manuel Blum, and Michael Shub. A simple unpredictable pseudo-random number generator, SIAM Journal on Computing, volume 15, pages 364–383, May 1986.

[23] B. Schneier,. Applied Cryptography. New York: Wiley, 1996.

[24] D. Johnson, A. Menezes. The Elliptic curve digital signature algorithm (ECDSA)”, February 24, 2000.

TABLE II

Performance comparison of our protocol to Flat, LKH and our previous protocol is given for five major criteria. Avg. rekeying workload and number of keys in satellite layer (SL) only applicable to our protocol for GEO and LEO-MEO SL.

$N \geq 10^6,\ r \approx 10^5,\ l = (500 - 1000),\quad n_l = N/n_s \geq 2048, n_s \approx 100,\ m_2 > m_1,\ q = l/n_s$

	Avg. rekeying workload for SL	Avg. # keys stored in SL	Avg. rekeying workload for TU	Avg. # keys stored in TU	Avg. # keys stored in member
Flat	$N{\cdot}r$	N	-	-	1
LKH	$(k \log_k N){\cdot}r$	N	-	-	$\log_k N$
Previous Protocol	$(k \log_k l)/m_1$	l	$\log_k n_l$	$n_l + \log_k l$	$\log_k n_l$
Proposed Protocol	$(klog_k q)/m_2$	$6n_l + l + 1 \approx 2l$	$\log_k n_l$	$n_l + \log_k l$	$\log_k n_l$
GEO	$k \log_k n_s \leq 10$	$6n_l + l + 1 \approx 2l$			
MEO-LEO	$(klog_k q)/m_2$	$q < l$			

TABLE III

Comparison of cryptographic protocols with regard to nine essential critera. RSA-S denotes RSA Signatures and DSA-V denotes DSA Variants

	DH	ECDH	RSA-S & DSA-V	ECPVSS	IMC Based	ECMQV
Authentication	No	No	Yes	Yes	Yes	Yes
Unforgeability	No	No	Yes	Yes	Yes	Yes
Integrity	No	No	Yes	Yes	Yes	Yes
KK-S	No	No	-	-	Yes	Yes
FS	No	No	-	-	Yes	Yes
KCI-R	No	No	-	-	Yes	Yes
BW Efficiency	M	H	L	VH	VL	H
Computational Efficiency	M	H	M	VH	M	H
Confidentiality	H	H	H	H	VH	H